CIRCUIT AND METHOD FOR PROVIDING A PRECISE CLOCK

Field of the Invention

The present invention relates to a method and/or

5 architecture for providing a precise clock generally and, more

particularly, to a method and/or architecture for providing a

precise clock without a precision timing element.

Background of the Invention

10 Computer, microcontroller and microprocessor systems use

one or more clock oscillator circuits to provide timing reference

signals. Conventional clock oscillator circuits using an external-

based clock can be used to provide a clock signal. A precision

timing component (i.e., crystal, ceramic resonator, etc.) that is

15 external to the oscillator circuit is required in an external-based

clock oscillator. The precision timing component supplies a

reference frequency for the clock oscillator. Interfacing to the

timing component requires one or two pins of the chip that the

external-based clock oscillator circuit is on. The precision

20 timing component adds to the cost of a system. The chip uses the

1

reference frequency presented by the timing component or further tunes the frequency to match the data rate of incoming data or a multiple of the data rate. Phase Lock Loop (PLL) and/or Delay Lock Loop (DLL) circuits can be used for locking to or matching data

5   rates. Locking to or matching data rates sometimes requires long data 'training' sequences. Schemes that use a local fixed reference frequency do some phase shifting or phase selection to sample incoming data since the external and internal clock rates differ.

10   Conventional Universal Serial Bus (USB) communication systems are PLL based, with an accurate local clock. The USB device can be compliant with the USB specification version 2.0 (published April 2000), which is hereby incorporated by reference in its entirety (see http://www.usb.org/developers/usb20/). USB

15   version 2.0 host Full-speed communications packets include accurately spaced 'Start Of Frame' (SOF) packets that arrive periodically every 1 ms ±0.05% (i.e., 1.000 ms ± 500 ns)(see USB specification version 2.0, sections 7.1.11 and 7.1.12).

A clock oscillator that takes advantage of the accurately

20   spaced USB SOF packets to calibrate the frequency of the clock signal would be desirable.

Summary of the Invention

The present invention concerns an apparatus comprising a circuit configured to (i) generate an output having a frequency and (ii) adjust the frequency in response to a measured duration of a

5   known time interval associated with an input data stream.

The objects, features and advantages of the present invention include providing a method and/or architecture for providing a precise clock that may (i) tune an oscillator to match a clock rate of a host system, (ii) require no external precision

10   timing component, (iii) require no chip pins, (iv) precisely lock to an incoming data rate with 0.2% precision or better, and/or (v) perform 0.2% accurate timing of data communications without an external timing component.

15   **Brief Description of the Drawings**

These and other objects, features and advantages of the present invention will be apparent from the following detailed description and the appended claims and drawings in which:

FIG. 1 is a block diagram illustrating a preferred

20   embodiment of the present invention;

3

FIG. 2 is a detailed block diagram illustrating a preferred embodiment of the present invention;

FIG. 3 is a timing diagram illustrating an example of a packetized communication protocol;

5   FIG. 4 is a timing diagram illustrating an example bit pattern of a start of frame packet of FIG. 3; and

FIG. 5 is a table illustrating an accuracy analysis of various oscillators implemented in accordance with the present invention.

10

Detailed Description of the Preferred Embodiments

Referring to FIG. 1, a block diagram of a circuit 100 is shown in accordance with a preferred embodiment of the present invention. The circuit 100 may be a self-calibrating oscillator
15   circuit. The circuit 100 may be used to generate a precise clock signal that may be, in one example, tuned to within 0.2% of a data rate of a data stream or a multiple of the data rate. The circuit 100 may be configured to maintain a precise clock frequency without a precision timing component (e.g., a crystal, ceramic resonator,
20   etc.). The circuit 100 may be configured to adjust a frequency of oscillation of a local oscillator in response to the data rate of

a data stream from a host system. The frequency of oscillation may be adjusted to within 0.2% of the host data rate. In one example, the data stream may be a USB data stream.

In general, the circuit 100 may generate a clock signal having an accuracy related to the host system data rate accuracy. The circuit 100 may have an input 102 that may receive the data stream (e.g., DATA) from the host system and an output 104 that may present a clock signal (e.g., CLK) that may be used, in one example, for data recovery. The signal DATA may be a serial data stream, for example, a full-speed communications data stream in accordance with the Universal Serial Bus (USB) version 2.0 standard (published April 2000), which is hereby incorporated by reference in its entirety. The signal DATA may have one or more periodic signal components of a known duration that may occur at precisely known time intervals. For example, USB version 2.0 start-of-frame (SOF) packets or any other appropriate periodic component that may be contained in a communication protocol.

The circuit 100 may comprise a circuit 106 and a circuit 108. The circuit 106 may be a programmable oscillator circuit. The circuit 108 may be a calibration circuit. The circuit 106 may have an input 110 that may receive a signal (e.g., CN) and an

output 112 that may present the signal CLK. The circuit 106 may be configured to (i) generate the signal CLK having a default (untuned) frequency and (ii) adjust the frequency of the signal CLK in response to the signal CN. The signal CN may be, in one

5   example, a digital tuning signal. However, other types of signals may be implemented accordingly to meet the design criteria of a particular application. The signal CN may be, in one example, 8-bits wide. However, other numbers of bit may be implemented to meet the design criteria of a particular application.

10   The circuit 108 may have the input 102 that may receive the signal DATA, an input 114 that may receive the signal CLK, and an output 116 that may present a signal CN. The circuit 108 may be configured to generate the signal CN in response to the signals DATA and CLK. For example, the signal CN may be generated in

15   response to a measurement of the signal DATA made with the signal CLK.

Referring to FIG. 2, a detailed block diagram of the circuit 100 is shown. The circuit 108 may comprise, in one example, a circuit 120, a circuit 122, and a circuit 124. The

20   circuit 120 may be a detector circuit. The circuit 122 may be a counter circuit. The circuit 124 may be a control circuit. The

signal DATA may be presented to an input of the circuit 120. The circuit 120 may have an output 126 that may present a signal (e.g., DET) to an input 128 of the circuit 122. The circuit 120 may be configured to generate the signal DET in response to precise

5   periodic events having a known time interval in the signal DATA. For example, the circuit 120 may be implemented as a SOF packet detector in accordance with the USB 2.0 standard. The signal DET may be a detection signal that may indicate, in one example, recognition of a start of frame packet. In one example, the signal

10   DET may comprise a pulse for each SOF packet detected. For example, when the signal DATA comprises USB 2.0 SOF packets from a host system, the interval between pulses of the signal DET may be 1 ms ± 500 ns. However, other intervals may be implemented for other communication protocols.

15   The circuit 122 may be implemented, in one example, as one or more counters. The circuit 122 may have an input 130 that may receive the signal CLK and an output 132 that may present a signal (e.g., CNT) to an input 134 of the circuit 124. The circuit 122 may be configured to measure a time interval delimited by the

20   signal DET. In one example, the circuit 122 may start counting transitions of the signal CLK in response to a first transition of

7

the signal DET and stop counting transitions of the signal CLK in response to a second transition of the signal DET. The circuit 122 may be configured, in one example, to present a measured value for the time interval as the signal CNT. In one example, the signal

5   CNT may comprise a count of a number of cycles of the signal CLK that occurred between two SOF packets in the signal DATA. Alternatively, the signal CNT may comprise a count of a number of cycles of the signal CLK that occur over a longer interval. For example, an interval comprising two or more SOF packets in the

10 signal DATA.

The circuit 124 may have an input 136 that may receive the signal CLK. The circuit 124 may be configured to generate the signal CN in response to the signals CNT and CLK. The circuit 124 may be programmed with the known interval between precise periodic

15 events of the signal DATA (e.g., the 1 ms ± 500 ns interval between USB 2.0 SOF packets). In one example, the circuit 124 may be configured to generate the signal CN as the difference between the known value for the interval and the measured value as presented by the signal CNT. The signal CN may be used to tune the frequency of

20 the circuit 106. In one example, the circuit 106 may be a digitally tunable oscillator circuit and the signal CN may comprise

digital tuning values from a look-up table addressed in response to

the difference between the known value and the measured value for

the time interval between, for example, SOF packets. In another

example, the digital tuning values for the signal CN may be

5   computed using the known value and the measured value for the time

interval. For example, when a difference between the known value

and the measured value for the time interval between SOF packets is

+7 counts off (e.g., the signal CNT = expected value +7) and the

oscillator 106 is configured to adjust 0.6 steps/count, the circuit

10   124 may compute a correction of -4.2 steps. A correction (tuning)

value of -4 may be presented to the oscillator 106 (e.g., CN = -4).

Referring to FIG. 3, a timing diagram illustrating a

packetized communications data stream is shown. The data stream

may contain a number of start of frame (SOF) packets 150 and a

15   number of data packets 152. The SOF and data packets may be

implemented in accordance with the USB 2.0 full-speed standard.

The USB specification generally requires that the time interval

between the SOF packets be maintained within 1.000 ms ± 500 ns.

The precise separation of the SOF packets 150 may be used as a

20   precision time reference (e.g., the variance of the time reference

may be guaranteed by the 0.05% data rate requirement for hosts).

When the programmable oscillator 106 is operating at a clock rate
of 24 MHz, 24,000 counts may occur in one frame interval. The
division of a frame interval into 24,000 parts (counts) may provide
sufficient resolution for tuning the local oscillator to within a
5   0.25% final accuracy, even under worst case conditions.

Referring to FIG. 4, a diagram illustrating an example
bit pattern of a SOF packet of FIG. 3 is shown. The occurrence of
the SOF packet in the data stream may be indicated by detection of
the SOF bit pattern. The SOF packet may be recognized without an
10   initially highly precise clock. The bit pattern of the SOF packet
generally contains no long runs of data without transitions. With
no long runs of data without transitions, the SOF packet generally
does not require an accurate clock for detection. The SOF packet
is generally recognizable within the first 15 bits (e.g., portion
15   200). In addition to the SOF identification bits (e.g., bits 7-
14), the SOF packet may comprise a number of sync bits (e.g., bits
0-6), a frame number (e.g., bits 15-25), a cyclic-redundancy-code
(CRC) (e.g., bits 26-30), possible bit stuffs, and an end-of-packet
(EOP) marker (e.g., bits 31-33). In one example, the bit patterns
20   may be presented in a Non-Return-to-Zero-Inverted (NRZI) format.

Referring to FIG. 5, a table illustrating an example of an accuracy analysis of the circuit 100 implemented with a target clock oscillator frequency of 24 MHz is shown. In accordance with the present invention, a clock oscillator may be tuned without an external precision timing component. In one example, a clock oscillator with an untuned accuracy of 10% may be adjusted to within 1.3% after one tuning, and within 0.26% after a second tuning (e.g., ~2 ms, or detection of three successive SOF packets). In another example, a clock oscillator with an untuned accuracy of 5% may be adjusted to within 0.7% after one tuning, and 0.19% after a second tuning. In general, a clock oscillator with an untuned accuracy within about 0.8% may be tuned to within 0.2% after one tuning (e.g., 1 ms or detection of two successive SOF packets). The data presented in FIG. 5 generally makes the following assumptions about the oscillator characteristics and error sources: (i) jitter in detecting successive SOF packets may be 175 ns (e.g., 2+ bit times); (ii) oscillator jitter may be 3ns; (iii) host SOF jitter may be 90 ns (e.g., 1 bit time) beyond the 500 ns of the USB specification; (iv) oscillator gain step size of 15 kHz/step at a 24 MHz clock rate; (v) gain step size may be trimmed to within 12%; and (vi) oscillator maximum linearity error may be 0.75 LSB. The

step size of 15 kHz may provide sufficient resolution in the

oscillator frequency. The above assumptions may be representative

of a preferred implementation of the invention. However,

acceptable performance to meet the design criteria of a particular

5   implementation may be achieved using different assumptions

concerning, for example, magnitude of error sources, gain step

size, etc. For example, at 24 MHz, a 0.2% error is generally 48

kHz. The 15 kHz step size may allow tuning within ±3 steps to meet

the USB specification. Higher frequency clocks may be used to

10   provide finer step sizes. The terminology used in FIG. 5 may be

summarized in the following TABLE 1:

TABLE 1

Field	Units	Description
Clock Err	%	This gives the initial accuracy of the internal clock before the tuning step. The value is relative to the incoming USB data traffic rate.
Frame Time	ns	= 1ms (host 0.05% error not accounted for – chip locks req becomes 0.20%)
Min Time Max Time	ns	Calculation of min (max) time for the total Bit Times, by subtracting (adding) Osc. Jitter and the appropriate Next/Pair Jitter to the nominal time.
ClockFreq	MHz	Nominal clock rate of internal oscillator. Initial error ('Clock Err') is from this number. This field, located above 'Min Count,' is entered by user.
Min Count Max Count	-	Calculation of min and max possible counts over the bit time window. Min = INT(MinTime * ClockFreq * (1 - ClockErr) / 1000) Max = INT(MaxTime * ClockFreq * (1 + ClockErr) / 1000) + 1
ClkMinRange ClkMaxRange	MHz	Calculation of possible frequency range in the oscillator that could give the Min or Max Counts in the counter over the given bit time range (between Min Time and Max Time). Note that the possible

12

		ranges for a given count can exceed the initial Clock Err inaccuracy, and reflects the number the circuit must use in determining the possible frequency. ClkMinRange = (Min Count - 1) * 1000 / (Max Time) ClkMaxRange = (Max Count + 1) * 1000 / (Min Time)
TuneAmount: AveMin AveMax	MHz	This gives the tune amount necessary, in this case for the worst case counts that could occur under the specified conditions. The amount is the difference between average of the ClkMinRange (ClkMaxRange) and the ideal frequency. TuneAmount = ClockFreq - (Average of ClkMin(Max)Range values)
TuneCounts MIN MAX	-	The MIN cases, shown in the second row, apply to AveMin TuneAmount. The MAX cases, shown laster in the second row, apply to the AveMax TuneAmount. This is the actual change in the oscillator trim setting, using the known step size for the oscillator. This is the amount of correction necessary to nominally trim the frequency to the right value.
MIN Tuning MAX Tuning	MHz %	These give the worst case final values of the clock after applying the correction. The error results from (A) the initial uncertainty in the clock frequency for a given count, (B) the error in the correction factor at this corner (Step Sizes Limit - nominal step size), and (C) Oscillator Linearity Error. The following two calculations illustrate a pair for the MIN case. Min Freq = ClkMinRange[smallest] + (TuneCount * Smallest StepSize) - (LinearityError * Smallest StepLimit) Max Freq = ClkMinRange[biggest] + (TuneCount * Biggest StepSize) + (LinearityError * Biggest StepLimit) The % value is just the % error from the ideal
Max Error… MIN MAX	%	This is just the maximum value from the preceding MIN and MAX tuning results.

In the USB system, the host generally transmits at a highly precise data rate (0.05% accurate) while 'Full Speed' (12 Mb/s data) devices are generally required to transmit with a

13

somewhat more relaxed precision (0.25%). The required precision typically requires a local timing element, such as a crystal, as the basis for accurate data rate transmission. In the present invention, no highly precise timing element is generally used. A reasonably stable oscillator circuit 106 may be present on chip, and generally requires an initial accuracy of only 5 to 10%. The oscillator 106 may be capable of fairly rapid tuning. The circuit 100 may be configured to time incoming data, for example, the interval between start-of-frame (SOF) packets. The SOF packets may be specified to be transmitted by the host at 1 ms intervals, with no more than a ±500 ns error. By timing the precision time interval, the circuit 100 may determine the error in the programmable clock oscillator 106, and adjust the oscillator accordingly.

With standard CMOS processing, the circuit 100 may quickly achieve a precise frequency within 0.2% of the host clock rate (using the SOF interval). The present invention may, in one example, exploit characteristics of USB traffic to achieve the required precision using only a single USB data frame. However, other numbers of frames may be used.

Following a reset, the circuit 100 will generally receive

a minimum of 10 SOF packets before transmitting anything.   The

circuit 100 may be configured to use the SOF interval to complete

the precision tuning of the circuit 106.   The SOF packets generally

5   arrive at 1 ms intervals.   The circuit 100 may be configured to

continuously fine tune the circuit 106 using subsequent intervals.

In one example, various options on counting may exist,

including:

1.   Choosing an appropriate clock rate;

10   2.   Choosing an initial wide-tuning range vs. a later limited

tuning range (to avoid erroneous behavior on a badly

placed SOF packet);

3.   Determining the method for recognizing the SOF; and/or

4.   Tuning the oscillator 106 directly on a single packet,

15   rather than the interval (e.g., 1ms) between 2 SOF

packets. However, tuning the oscillator 106 directly may

require a much higher frequency clock, for sufficient

oversampling.

An alternative embodiment of the present invention may

20   use pulses generated in response to the arrival of the SOF packets

as the reference signal of a phase locked loop (PLL). The PLL would generally lock to the SOF packet rate. However, the use of a PLL involved addition design to achieve the desired accuracy in the required time (e.g., ~10 ms).

5   The present invention may take advantage of generally not needing to transmit with a precise clock until after a number of packets that are accurately timed by the host are received. The circuit 100 may initially tune during a reset time after a bus reset. The reset time may be, in one example, at least 10 ms 10 (e.g., USB 2.0, section 7.1.7.6, time interval $\Delta t6$). Receiving 6 or fewer SOF packets will generally be sufficient to tune the oscillator 106 to within 0.25% in the accordance with the present invention. Following this initial tuning, continual incremental tuning may occur with each frame, or each pair of frames.

15   In one example, the oscillator 106 may need to be trimmed to provide a reasonably accurate frequency step size (e.g., a "gain" trim). Once the trimming is done, an "offset" trim to close to the target frequency may also be done. The trim values may be stored in non-volatile memory (e.g., EPROM, EEPROM, flash memory, 20 etc.). The trim values stored in non-volatile memory may be loaded

into the oscillator 106 during initialization (e.g., boot-loaded).

For parts without non-volatile memory, a laser trimming step may be

used.

The following process may illustrate an example operation

5   of a device implemented in accordance with the present invention.

Step 1.   The device may power up at a reset oscillator

frequency.   In general, the reset frequency may be

less than the final target frequency.

Step 2.   The device may be automatically brought close to

10   the target frequency via the factory-set trim.   In

one example, the factory-set trim may be loaded

from a non-volatile memory (e.g., EPROM, flash

memory, EEPROM, etc.) and presented through a boot-

loader during initialization.   In another example,

15   the factory-set trim may already be available

through a laser trim.   The trim setting may set the

oscillator gain within ~10% of the ideal value, and

the actual frequency to within 5% of the final

frequency.

17

Step 3.   A SOF interval counter may be reset.   The reset

value may be, in one example, zero.   However, any

other convenient value (e.g., 576 (240 hex)) may be

used to simplify tuning, as explained below).

5   Step 4.   A SOF packet may arrive, and may be signaled by a

SOF detector.   In one example, a standard USB

Serial   Interface   Engine   (SIE)   may   be   used.

However, other appropriate detection circuits may

be implemented to meet the design criteria of a

10   particular application.   The arrival of the SOF

packet may start the interval counter.

Step 5.   Another SOF packet may arrive and may be again

signaled by the SOF detector, stopping the interval

counter.   The count value will generally be 24000

15   counts from the reset value, for a clock frequency

of 24 MHz.   Deviation from the 24000 target count

generally indicates an internal clock mismatch to

the system clock.

Step 6.   The circuit 100 may use the count error (e.g., the

20   difference from the expected and the actual count)

18

to look up a frequency correction value from a look-up table. In another example, the frequency correction value may be computed from the count error. The correction value may be applied to the on-chip digitally-controlled oscillator 106.

Step 7.   The counter(s) may be reset. The next pair of SOF packets may again start and stop the counter(s), and a new correction number may be looked up (or computed) and applied to the oscillator.

Step 8.   The step 7. may be repeated as often as desired, possibly on every pair of SOF packets.

Several options may be used for optimizing the circuit 100, including but not limited to:

(i) Using two different tuning modes – one for the initial tuning from power up, and another applied once the oscillator is 'locked.' The initial phase may allow wider tuning ranges, while in the stable phase the increment amount may be limited to prevent rapid changes in frequency.

(ii) Returning initially to the wider tuning mode after leaving a suspend mode.

(iii) Determining the limit, if any, for aborting the tuning on a count error that is outside a pre-defined acceptable range.

(iv) Tuning continuously from one SOF to the next, vs. tuning on SOF pairs. In the a first method, tuning may occur from SOF #1 to SOF #2, then SOF #2 to SOF #3, then SOF #3 to SOF #4, etc. In a second method, tuning may occur from SOF #1 to SOF #2, then from SOF #3 to SOF #4, then from SOF #5 to SOF #6, etc. The first method may provide more tuning events over a given time interval, but generally at the expense of a more complex counter design. In the second method, the counter may be stopped at the second SOF of each pair, and the result used for the tuning. In the first method, the counter result may be, in one example, captured (e.g., in a latch, register, etc.) and then immediately reset for the next interval. Alternatively, two counters may be used alternately to count successive intervals.

(v) Tuning across longer time intervals than between two successive SOFs. The counting interval may be chosen to be any integral number of SOFs, such as measuring across two frame times (e.g. from SOF #1 to SOF #3 or from SOF #n to SOF #n+2, where n is an integer). Generally, the interval may be measured across a number of SOF packets, for example, measuring from SOF #n to SOF #n+m where n and m are integers.

When the circuit 106 is updated, a faster-than-desired clock may occur for one cycle. To prevent updates to the digitally controlled oscillator from causing a somewhat faster-than-desired clock for one cycle at the update, the circuit 100 may be configured to filter or gate off the clock for one cycle during updates.

In one example, the SOF detect circuit 120 may only need to follow a SOF packet through a packet identifier field (PID). Following the packet only through the (PID) may relax the timing accuracy requirements (no long runs without a transition).

In one example (e.g., for a 24 MHz clock and a 1ms interval), the number of counts in a 1ms interval will nominally be

24000. Resetting the counter 122 with a value of 576 may result in the LSBs being zero under the ideal case (e.g., count=24576 = 0x3000hex). By resetting the counter to 576, a number of the LSBs may be used as an index into a correction look-up table. An

5 example look-up table may be summarized in the following TABLE 2:

TABLE 2

For |Count-24000| in range of:		
Minimum	Maximum	Tuning Amount (steps) [Tune up if count <24000 Tune down if count >24000]
0	7	0
8	22	1
23	37	2
8+15*(N-1)	7+15*N	N

Table 2 assumes a 24 MHz clock counting over 1 ms interval and oscillator gain = 15 kHz/step. However, other appropriate look-up table values may be selected to meet the design criteria of the particular design.

20 When the signal CN is selected to have 8 bits and a step size is selected to be ~15kHz, the oscillator circuit 106 may have a range of about 4 MHz. However, other step sizes for the oscillator 106 and other numbers of bits for the signal CN may be

0325.00417
CD00111

implemented to meet the design criteria of a particular application.

While the invention has been particularly shown and described with reference to the preferred embodiments thereof, it will be understood by those skilled in the art that various changes in form and details may be made without departing from the spirit and scope of the invention. For example, while the present invention has been described with respect to the full speed mode of the Universal Serial Bus, the present invention may be applicable to other data communications protocols as well.